Assignment 5-3 Palindromes

A palindrome is a number that reads the same backward as forward. For example, each of the following positive integers is a palindrome: 12321, 55555, 45554 and 11611. The problem is to find out whether the given integer is a palindrome.

Input

The input consists of t ($30 \le t \le 40$) test cases. The first line of the input contains only positive integer t. Then t test cases follow. Each test case consists of exactly one line with a positive integer n which is less than 2^{31}.

Output

For each such integer n, you are to output a single line containing the word "palindrome" or "non-palindrome" depending on whether the integer n is a palindrome.

Sample Input

2
12345
12321

Sample Output

non-palindrome
palindrome

Requirements

You are required to write a recursive function `bool isPalindrome( int first, int last )` to complete the following program which solves this problem. In this program `digits[ i ]` is the i-th digit of `number`. The function `isPalindrome` returns true if and only if `digits[ first ] == digits[ last ] && digits[ first + 1 ] == digits[ last - 1 ] && digits[ first + 2 ] == digits[ last - 2 ]` ⋯.

```
#include <iostream>
using namespace std;

// returns true iff the subarray digits[ first .. last ] is a palindrome
bool isPalindrome( int first, int last );

int digits[ 12 ] = {};

int main()
{
```

```
   int numCases;
   cin >> numCases;
   for( int i = 1; i <= numCases; i++ )
   {
      int n;
      cin >> n;

      int length = 0;
      for( int i = n; i > 0; i /= 10 )
         digits[ length++ ] = i % 10;

      if( isPalindrome( 0, length - 1 ) )
         cout << "palindrome" << endl;
      else
         cout << "non-palindrome" << endl;
   }
}

bool isPalindrome( int first, int last )
{




}
```